Ex6 :procedure

Objective

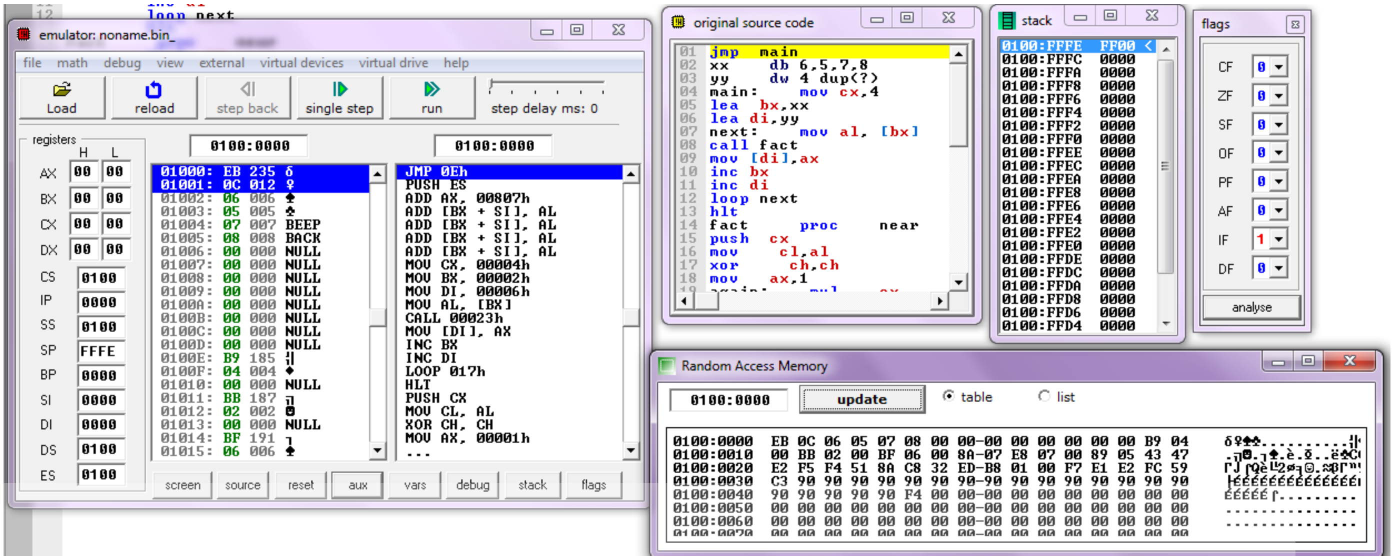
The objective of this experiment is to:

- **Implement a procedure (to compute the FACTORIAL of 6, 5, 7 and 8).**

Procedure

1. Study, type, emulate and run the following program step by step (recording the contents of all affected registers, flags, memory and the stack throughout all steps):

```
         jmp  main
xx       db 6,5,7,8
yy       dw 4 dup(?)
main:    mov cx,4
         lea  bx,xx
         lea di,yy
next:    mov al, [bx]
         call fact
         mov [di],ax
         inc bx
         inc di
         loop next
         hlt
fact     proc    near
         push  cx
         mov    cl,al
         xor    ch,ch
         mov   ax,1
again:   mul    cx
         loop   again
         pop  cx
         ret
fact     endp
```



2. Change the value 8 in xx to 9. Emulate and run the program (press run) while
watching the memory content at yy.

Report:

1. Discuss why xx= 9 gives a wrong result in the above factorial procedure.

   Because BX cannot contain more than 4 hexadecimal numbers and any number greater than 8 will need more than 4 digits in hexadecimal

2. Discuss: Why CX is pushed to the stack at the beginning of the procedure?

Because then CL and CH will change, so we pushed it first to have its value

3. Try to write and implement a procedure that computes the value of any term (up
   to 12 terms) in

   Fibonacci Sequence and return result in AX:

$$F(n) = F(n-1) + F(n-2)$$

Done by: Abeer(www.UOB-BH.com)

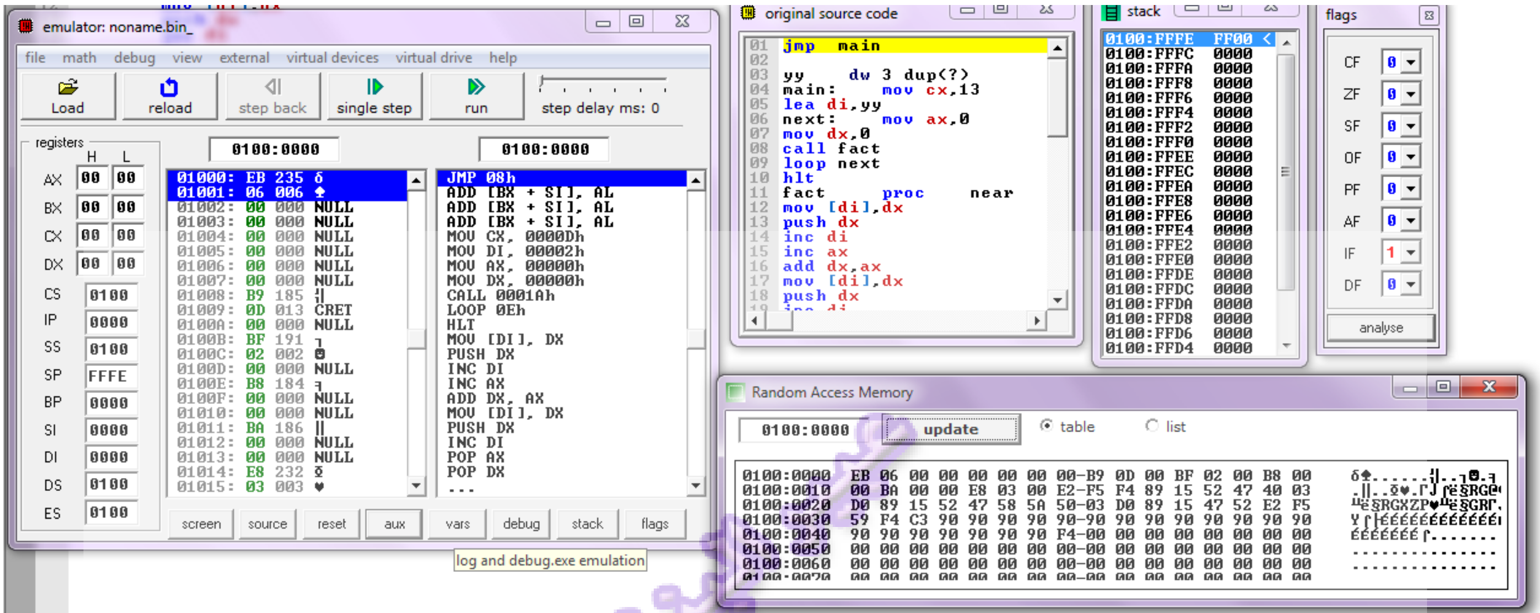
where F (0) = 0 & F (1) = 1

that is, F (2) = F (1) + F (0) = 0 + 1 = 1
F (3) = F (2) + F (1) = 1 + 1 = 2
and so on
The sequence of numbers (in decimal) will look like:

Z (n) = 0 1 1 2 3 5 8 13 21 . . .
Where term(0) = 0, term(1) = 1, term(7) = 13 ……etc

```
jmp  main
     yy       dw 3 dup(?)
main:         mov cx,13
lea di,yy
next:         mov ax,0
mov dx,0
call fact
loop next
hlt
fact     proc     near
mov [di],dx
push dx
inc di
inc ax
add dx,ax
mov [di],dx
push dx
inc di
again:        pop ax
pop dx
push ax
add dx,ax
mov [di],dx
inc di
push dx
loop again
pop cx
```

hlt
ret
fact endp



Done by: Abeer(www.UOB-BH.com)